UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/720,042	11/21/2003	Jacob Strom	8196-16	2544

20792 7590 11/25/2005

MYERS BIGEL SIBLEY & SAJOVEC
PO BOX 37428
RALEIGH, NC 27627

EXAMINER	
BROOME, SAID A	
ART UNIT	**PAPER NUMBER**
2671	

DATE MAILED: 11/25/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/720,042	STROM ET AL.	
	Examiner	Art Unit	
	Said Broome	2671	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 December 2002*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-28* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-28* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 3, 6, 17, 25 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Olsen et al.(US Patent 6,094,2000) in view of Wood et al.(US Patent 6,204,856).

Olsen et al. teaches all the limitations of claims 1 and 23 except for defining a plurality of rows of tiles comprising a plurality of rows of pixels with each tile including at least two rows of pixels. Regarding the preamble of claim 1, Olsen teaches a graphics processing method in column 2 lines 34-37, and is also illustrated in Figure 4 as element 40. Regarding the preamble of claim 23, Olsen et al. teaches a computer graphics system that determines occlusion for graphics primitives, which would contain computer program code embodied on a computer readable medium in order to execute the occlusion computations. Regarding claims 1 and 23, Olsen et al. teaches setting occlusion flags for respective tiles of a row of tiles, or pixels comprising the region for a graphics primitive, and whether the respective representative depth values meet an occlusion criterion in column 5 lines 52-56 where it its described that the depth values of the region comprising the pixels from the graphics primitive are compared and a determination of occlusion is then determined based on the result from the depth compare result register, illustrated in Figure 3 as element 35. Olsen et al. also teaches processing pixels in rows

of pixels in a row-by-row manner responsive to the occlusion flags in column 5 lines 22-23 where it is described that the depth values of each pixel of the graphics primitive, which contains rows of tiles, are processed, therefore each row would be processed as each pixel comprising the rows would be processed. Again Olsen et al. fails to teach defining a plurality of rows of tiles comprising a plurality of rows of pixels with each tile including at least two rows of pixels. Wood et al. teaches a defined plurality of blocks of tiles comprising rows of pixels in column 1 lines 6-10, in which each tile includes pixels from at least two rows of pixels, as described in column 4 lines 42-44 and is also illustrated in Figure 2. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al. with Wood et al. because this combination would provide the determination of occlusion of each row of pixels comprising the tiles of a graphics primitive, which would reduce computation time by providing only the visible pixels to be rendered.

Regarding claim 3, 17 and 25, Olsen et al. fails to teach processing rows of pixels using a zig-zag traversal algorithm. Wood et al. teaches processing pixels contained in the rows of tiles using a zig-zag traversal algorithm in column 5 lines 15-20. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al. with Wood et al. because this combination would provide an efficient algorithm that accurately processing the rows of pixels within a tile.

Regarding claim 6, Olsen et al. teaches a color buffer that stores respective color values in column 2 lines 62-64. Olsen et al. also teaches storing a color values and a depth value for the graphics primitive for the pixel in the color buffer and the depth buffer respectively in column 2 lines 62-64.

Regarding claim 26, Regarding the preamble of claim 26, Olsen et al. teaches a computer graphics system that determines occlusion for graphics primitives, which would contain computer program code embodied on a computer readable medium in order to execute the occlusion computations. Wood et al. teaches a defined plurality of blocks of tiles comprising rows of pixels in column 1 lines 6-10, in which each tile includes pixels from at least two rows of pixels, as described in column 4 lines 42-44 and is also illustrated in Figure 2. Olsen et al. teaches setting the occlusion flag which indicates non-occlusion in column 5 lines 42-45, where it is described that the depth compare result register is set to a value of 1 to indicate non-occlusion of the pixels which are known to comprise a tile of pixels. Olsen et al. also teaches detecting a flag, or logic value, indicating non-occlusion in column 6 lines 8-13. Olsen et al. also describes processing a pixel for the graphics primitive in steps 42 and 43 of Figure 4, where it is illustrated that the pixels defining the graphics primitive are processed.

Claims 2, 4, 5, 7-16, 18, 20-24, 27 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Olsen et al.(US Patent 6,094,2000) in view of Wood et al.(US Patent 6,204,856) in further view of Larson (US Patent 6,313,839).

Regarding claim 2, Olsen et al. teaches processing pixels responsive to occlusion flag, which is equivalent to the depth compare result register, in column 6 lines 7-12. Olsen et al. and Wood et al. fail to teach a first tile, or region, and a second tile for processing. Larson et al. teaches a first and second region, or tile, for processing in column 6 lines 8-10. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide faster processing of occlusion detection

for multiple tiles, which would enable quicker rendering of the visible pixels of graphics primitives.

Regarding claims 4 and 22, Olsen et al. teaches in column 6 lines 8-13, setting an occlusion flag, or depth compare result register, for respective pixels. Olsen and Wood et al. fail to teach occlusion flags and respective occlusion threshold depth values stored in a tile occlusion information cache, determining a maximum depth value and comparing the maximum depth value to the cached occlusion threshold depth value. Larson teaches storing occlusion threshold depth values in a cache memory in column 5 lines 63-66, and determining a maximum depth value for the graphics primitive in column 1 lines 59-60. Larson also teaches comparing the maximum depth value with the cached occlusion threshold depth value in column 2 lines 46-51. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olson et al., Wood et al. and Larson because this combination would provide decreased processing time of pixel of the graphics primitive during determination of occlusion by storing occlusion flags in a cache memory.

Regarding claim 5, Olsen et al. and Wood et al. teach all the limitations except a depth buffer that stores respective occlusion threshold depth values. Olsen et al. teaches setting the occlusion flag which indicates non-occlusion in column 5 lines 42-45, where it is described that the depth compare result register is set to a value of 1 to indicate non-occlusion of the pixels which are known to comprise a tile of pixels. Olsen et al. also teaches detecting a flag, or logic value, indicating non-occlusion in column 6 lines 8-13. Olsen et al. also describes processing a pixel for the graphics primitive in the tile without retrieving an occlusion threshold depth value for the pixel from the depth buffer in steps 42 and 43 of Figure 4, where it is illustrated that the

pixels defining the graphics primitive are processed without referencing an occlusion threshold depth value. Again, Olsen et al. and Wood et al. fail to teach a depth buffer that stores respective occlusion threshold depth values. Larson teaches a z buffer, or depth buffer, which stores the occlusion threshold depth values, or maximum and minimum z values, in column 1 lines 55-57. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide efficient access to occlusion threshold values by storing them into a depth buffer.

Regarding claim 7, Olsen et al. and Wood et al. teach all the limitations except for comparing the determined depth value to the occlusion threshold depth value and updating the occlusion threshold depth value in the tile occlusion information threshold cache to the determined depth value. Olsen et al. teaches determining a depth value for the graphics primitive for the pixel in column 3 lines 53-57 where it is described that pixel data is determined including the depth value. Again, Olsen et al. fails to teach comparing the determined depth value to the occlusion threshold depth value and updating the occlusion threshold depth value in the tile occlusion information threshold cache to the determined depth value. Larson teaches comparing the determined depth value to the occlusion threshold depth value in the tile occlusion information cache in column 2 lines 47-51, where it is described that the z value, or determined depth value, is compared to the maximum z value, or occlusion threshold depth value, that is stored in the cache memory. Larson also teaches updating the occlusion threshold depth value in response to the comparison to the determined depth value in column 2 lines 52-61, where it is described that the z value stored in the cache memory is updated and replaced by the determined

depth value that was compared. The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 4.

Regarding claim 8, Olsen et al. and Wood et al. fail to teach the claimed limitations. Larson teaches an aggregate tile occlusion information memory configured to store respective occlusion threshold depth values in column 4 lines 30-34, where it is described that a memory area stores the minimum and maximum depth values, or occlusion threshold depth values, within each region, which is equivalent to a tile comprising a block of pixels consisting of rows. Larson also teaches reading out z max and z min, which are occlusion threshold depth values, from the z limit buffer, which is stored in memory, and storing them into the cache memory. Larson teaches updating the z min, or occlusion threshold depth value, in the cache memory in column 6 lines 23-33. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide efficient storage of depth values for a region, which would then be utilized to determine and to responsively update occlusion properties for pixels within that region.

Regarding claim 9, Olsen et al. and Wood et al. teach all the limitations except detecting that the tile, or region of rows of pixels, has an occlusion flag indicating possible occlusion and updating the occlusion threshold depth value for the tile in the tile occlusion information cache responsive to the written z-value of the pixel. Olsen et al. teaches setting an occlusion flag, or depth compare result register, to indicate possible occlusion in column 6 lines 8-13. Again, Olsen et al. and Wood et al. fail to teach detecting that the tile, or region of rows of pixels, has an occlusion flag indicating possible occlusion and updating the occlusion threshold depth value for the tile in the tile occlusion information cache responsive to the written z-value of the pixel.

Larson teaches updating the occlusion threshold depth value, or min and max z value, in the cache memory responsive to the written z-value of the pixel in column 3 lines 9-18, where it is described that the min and max z values are updated and the received z value is written to a location in the cache memory. Larson describes comparing a depth value to an occlusion threshold depth value for the pixel in the depth buffer responsive to the determination of occlusion in column 4 lines 54-66. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide a reduction of processing time used to store and compare z values of row of pixels by storing the occlusion flags in a cache memory thereby ensuring quick and accurate determination of occluded regions.

Regarding claim 10, Olsen et al. and Wood et al. teach all the limitations except updating the occlusion threshold depth value for the tile in the tile occlusion information cache responsive to the written z-value of the pixel. Olsen et al. teaches a color buffer that stores respective color values in column 2 lines 62-64. Olsen et al. also teaches storing a color values and a depth value for the graphics primitive for the pixel in the color buffer and the depth buffer respectively in column 2 lines 62-64. Olsen et al. teaches setting the occlusion flag which indicates non-occlusion in column 5 lines 42-45, where it is described that the depth compare result register is set to a value of 1 to indicate non-occlusion of the pixels which are known to comprise a tile of pixels. Again, Olsen et al. and Wood et al. fail to teach updating the occlusion threshold depth value for the tile in the tile occlusion information cache responsive to the written z-value of the pixel. Larson teaches updating the occlusion threshold depth value, or min and max z value, in the cache memory responsive to the written z-value of the pixel in column 3 lines 9-18, where it

is described that the min and max z values are updated and the received z value is written to a location in the cache memory. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide efficient access to occlusion threshold values and color values by storing them into a depth and color buffers respectively.

Regarding claims 11 and 13, Olsen et al. and Wood et al. teach all the limitations except determining whether a second row of pixels is in a first row of tiles. Regarding the preamble of claims 11 and 13, though Larson does not explicitly teach storing occlusion flags in a cache memory, the reference does teach storing z values in a cache memory in column 2 lines 47-51 where it is described that a determination of whether the tested z value is occluded, therefore it would be obvious to store the result of the determination as an occlusion flag in the cache as well. Regarding claims 11 and 13, Larson teaches processing specific regions, or tiles, of pixels in column 6 lines 8-10, therefore it would be a minor computation to process the first row of pixels specifically, which contain z values stored in a cache memory, as described in column 2 lines 47-51. Though Larson does not explicitly teach storing occlusion and status flags, the reference does teach the determination of occlusion for z values in the region in column 2 lines 47-51, therefore it would be obvious to indicate occlusion based on the determined result using a flag or other indicator. Regarding claims 11 and 13, Larson teaches separately processing different regions, or tiles, of pixels to test the z values for occlusion in column 6 lines 5-10, therefore it would have been an obvious modification to determine whether a pixels from a second row is within a first row of tiles. Regarding claim 11, Larson also teaches independently processing a first and second region, or tile in column 6 lines 8-10 which contain all the rows of

pixels, using data from the cache memory in column 6 lines 5-18, therefore it would have been a minor modification to obtain information from a first row of pixels if the second row of pixels is contained the first row of tiles. Regarding claim 13, Larson teaches independently processing regions, or tiles, in column 6 lines 8-10 which contain z values that are tested to determine occlusion and stored in a cache memory as described in column 2 lines 47-51, therefore it would be an obvious modification to write, load and process any depth value responsive to the region and to store the values into the memory comprising tile information, as described in column4 lines 29-34, or the cache memory. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide a reduction of processing time used to store and compare z values of row of pixels by storing the occlusion flags in a cache memory thereby ensuring timely and accurate determination of occluded regions.

Regarding claims 12 and 13, Olsen et al. and Wood et al. teach all the limitations except establishing an aggregate tile occlusion information memory to store respective occlusion threshold depth values for all tiles and storing occlusion threshold depth values for the first row of tiles from the aggregate tile occlusion information memory in the tile occlusion information cache. Olsen et al. teaches setting the occlusion and status flags in the memory to predetermined values in column 4 lines 63-64, where it is described that when the depth compare result register, which is equivalent to the occlusion flag, is initialized in step 41 of Figure 4 is has a predetermined logic value, or status flag. Again, Olsen et al. and Wood et al. fail to teach establishing an aggregate tile occlusion information memory to store respective occlusion threshold depth values for all tiles and storing occlusion threshold depth values for the first row

of tiles from the aggregate tile occlusion information memory in the tile occlusion information cache. Regarding claims 12 and 13, Larson teaches an aggregate tile occlusion information memory configured to store respective occlusion threshold depth values in column 4 lines 30-34, where it is described that a memory area stores the minimum and maximum depth values, or occlusion threshold depth values, within each region, which is tile comprising a block of pixels consisting of rows. Larson also teaches storing occlusion threshold depth values, or min and max z values, from a first region(column 6 lines 8-9), from the aggregate tile occlusion information memory, or RAM as describe din column 4 lines 30-34, into the tile occlusion information cache, or cache memory as described in column 6 lines 23-30. The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 11.

Regarding claim 14, Olsen et al. and Wood et al. fail to teach the limitations. Larson teaches separately processing different regions, or tiles, of pixels to test the z values for occlusion in column 6 lines 5-10, therefore it would have been an obvious modification to determine particular locations of rows and whether a pixels from a second row is within a first row of tiles. Larson also teaches updating the occlusion threshold depth value in response to the comparison to the determined depth value in column 2 lines 52-61, where it is described that the z value stored in the cache memory is updated and replaced by the determined depth value that was compared. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide efficient storage of depth values for a region, which would then be utilized to determine and to responsively update occlusion properties for pixels within that region.

Regarding claim 15, Olsen et al. and Wood et al. teach all the limitations except a display and a graphics processor coupled to the display. Olsen et al. teaches setting an occlusion flag, or depth compare result register, to indicate possible occlusion in column 6 lines 8-13, where it is described that the register is set to a certain value indicating the visibility of the pixel under analysis. Olsen et al. also teaches processing pixels in rows of pixels in a row-by-row manner responsive to the occlusion flags in column 5 lines 22-23 where it is described that the depth values of each pixel of the graphics primitive, which contains rows of tiles, are processed, therefore each row would be processed as each pixel comprising the rows would be processed as well. Wood et al. describes a plurality of rows of tiles in a graphics display in column 4 lines 41-44. Wood et al. also describes each tile including pixels from at least two rows of pixels in column 4 line 44 where it is described that a tile could contain 8x8 pixels and therefore, based on those dimensions, contains at least two rows of pixels. Again, Olsen et al. and Wood et al. fail to teach a display and a graphics processor coupled to the display. Regarding the preamble of claim 15, Larson teaches an apparatus in column 1 lines 52-54. Regarding claim 15, Larson also teaches a display in column 3 lines 46-48, and it is also illustrated in Figure 1 as element 21. Larson also teaches a graphics processor coupled to the display in column 3 lines 57-59. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide an apparatus that determines occlusion of each row of pixels comprising the tiles for a graphics primitive, which would reduce computation time by providing the visible pixels to be rendered.

Regarding claim 16, Olsen et al. teaches processing a portion of pixels in a row of tiles, or region, responsive to the occlusion flag, or depth compare result register, in column in column

6 lines 7-12. Olsen et al. and Wood et al. fail to teach the graphics processor and a first and second tile for processing. Larson teaches a graphics processor in column 3 lines 57-59. Larson also describes a first and second region, or tile, in column 6 lines 8-10. The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 15.

Regarding claim 18, Olsen et al. and Wood et al. fail to teach the claimed limitations. Larson teaches determining a maximum depth value for the graphics primitive in column 1 lines 59-60. Larson also teaches that a depth value, or z value, is compared to determine if that value is less than an established non occluded minimum depth value in column 6 lines 23-30, therefore it would have been a minor modification to compare the maximum depth value with the minimum non occlusion depth value. The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 15.

Regarding claim 20, Olsen et al. and Wood et al. teach all the limitations except a display and a graphics processor coupled to the display. Olsen et al. teaches setting the occlusion flag which indicates non-occlusion in column 5 lines 42-45, where it is described that the depth compare result register is set to a value of 1 to indicate non-occlusion of the pixels which are known to comprise a tile of pixels. Olsen et al. also teaches detecting a flag, or logic value, indicating non-occlusion in column 6 lines 8-13. Olsen et al. also teaches processing a pixel for a graphics primitive in column 5 lines 22-23. Again, Olsen et al. and Wood et al. fail to teach a display and a graphics processor coupled to the display. Regarding the preamble of claim 20, Larson teaches an apparatus in column 1 lines 52-54. Larson also teaches a display in column 3 lines 46-48, and it is also illustrated in Figure 1 as element 21. Larson also teaches a graphics

processor coupled to the display in column 3 lines 57-59. The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 15.

Regarding claim 21, Olsen et al. teaches processing a pixel for the graphics primitive in the tile without retrieving an occlusion threshold depth value for the pixel from the depth buffer in steps 42 and 43 of Figure 4, where it is illustrated that the pixels defining the graphics primitive are processed without referencing an occlusion threshold depth value. Olsen et al. and Wood et al. fail to teach a graphics processor that maintains a depth buffer configured to store respective occlusion threshold depth values. Larson teaches a graphic processor in column 3 lines 57-59, which contains a z buffer, or depth buffer, which stores the occlusion threshold depth values in column 4 lines 35-39. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide efficient access to occlusion threshold values by storing them into a depth buffer.

Regarding claim 27, Olsen et al. also describes processing a pixel for the graphics primitive in the tile without retrieving an occlusion threshold depth value for the pixel from the depth buffer in steps 42 and 43 of Figure 4, where it is illustrated that the pixels defining the graphics primitive are processed without referencing an occlusion threshold depth value. Olsen et al. and Wood et al. fail to teach a depth buffer that stores respective occlusion threshold depth values. Larson teaches a z buffer, or depth buffer, which stores the occlusion threshold depth values, or maximum and minimum z values, in column 1 lines 55-57. The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 21.

Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over Olsen et al.(US Patent 6,094,2000) in view of Wood et al.(US Patent 6,204,856), in further view of Larson(US Patent 6,313,839) and in further view of Dye et al.(US Patent 6,518,965).

Regarding claim 19, Dye et al. teaches a display and graphics processor housed in a portable electronic device in column 6 lines 32-46. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al, Larson and Dye et al. because this combination would provide a more efficient use of the limited resources contained in portable electronic devices.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Said Broome whose telephone number is (571)272-2931. The examiner can normally be reached on 8:30am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ulka Chauhan can be reached on (571)272-7782. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

S. Broome 11/16/2005

11/22/05

RICHARD HJERPE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2600